A Multi-Hypothesis Constraint Network Optimizer for Maximum Likelihood Mapping

Dario Lodi Rizzini, Stefano Caselli
RIMLab - Robotics and Intelligent Machines Laboratory
Dipartimento di Ingegneria dell'Informazione
University of Parma, Italy
E-mail {dlr,caselli}@ce.unipr.it

***Abstract*— Loop closure is one of the most difficult task in localization and mapping problems since it suffers from perceptual aliasing. Multi-hypothesis topological SLAM algorithms have been developed to exploit connectivity and disambiguate such difficult task.**

In this paper, we propose a multi-hypothesis constraint network algorithm that tracks multiple map topologies and simultaneously keeps metric information. The map is stored as a graph consisting of poses and constraints and each constraint is associated to a loop closure hypothesis. Hypotheses are stored in a hypothesis tree that is expanded whenever possible loop closure may occur. Network poses are computed according to the most likely topological configuration, but alternative pose values are also computed for the poses that are adjacent to a hypothesis constraint to recover quickly the new configuration when required. Results provide a validation of the proposed approach.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is the problem of concurrently building a map from the sensor measurements acquired by a robot, while concurrently estimating the position in the map. Mapping has become an important task for several robotic applications including navigation and human-robot interaction. In literature, several methods, either based on Bayesian filtering or on maximum-likelihood (ML) estimation, have been proposed to address SLAM and mapping problems.

If map construction were achieved by integrating raw odometry and observations, the resulting map would be degraded by the uncertainty of measurements. To recover the degraded information mapping algorithms try to detect when a robot has returned to a previously visited region of the environment. The *loop closure* is a crucial operation for the estimation map since map metric error can be reduced by recovering the topological consistency. However, the identification of previously visited places can be difficult when multiple places appear indistinguishable through sensor observations. This *perceptual aliasing* can lead to wrong data associations and to the final corruption of estimated map.

Several data association techniques have been proposed to close the loop reliably. In feature-based maps loop closure is achieved by matching features extracted from raw observation with map landmarks. Individual comparison is commonly performed with validation gate based on *Mahalanobis distance* bewteen landmark and feature parameters. *Joint compatibility branch and bound* (JCBB) [1] and the Hungarian algorithm [2], [3] have been proposed to find the optimal assignment between two collections of features instead of nearest neighbor strategy. All these techniques require an approximated estimation of robot pose in order to match the local features in the map reference frame. *Combined constraint data association* (CCDA) [4] performs association by building the constraint graphs both of features and of map landmarks and searching maximum clique. In the view-based representation of the environment (see e.g. [5]) each observation is associated to the robot pose from which the observation has been acquired. Thus, loop closure is performed comparing the current view with the other local maps stored in the map and the outcome is a constraint between the two related poses. The method for estimating the likelihood of a candidate association depends on the sensor data category. Occupancy grid maps are usually compared using correlation techniques [6]. Visual appearence [7] is a popular method to close loops with vision sensors.

The methods listed before provide an estimation of loop closure using the current measurement. Map building algorithms usually applies a data association technique and, once a wrong association is estabished, the resulting map cannot be corrected. Since loop closure decisions based only on current measurement are difficult due to perceptual aliasing, the sequence of observations acquired during robot exploration can be exploited. In [8]) loop closure is established only after observing a sequence of images. Several methods track multiple outcomes of association by building an hypothesis tree. Dudek *et al* [9] pioneered the multi-hypothesis topological map approach using an exploration tree. In [10] the multi-hypothesis approach is compared with FastSLAM [11] and other methods that track multiple hypotheses and an algorithm for "repairing" data associations is described. Recently, a multi-hypothesis topological algorithm has been proposed to solve perceptual aliasing [12].

In this paper, we illustrate a multi-hypothesis maximum-likelihood mapping algorithm. The algorithm formulates the mapping problem as a graph of robot poses and constraints between poses encoding both metric and topological information [13]. Loop closing operation is performed by inserting in the graph an edge representing a spatial connection between the current pose and a previously visited one. Thus, each association hypothesis corresponds to a different graph topology like for other topological maps. However,

our aim is the estimation of the map configuration (i.e. the state vector of robot poses) that maximizes the likelihood function and there is a different likelihood function for each hypothesis. Since keeping separate graph maps for each hypothesis would result in a tremendous demand of both memory and computation, the graph is shared between different hypotheses. The edges inserted for a specific hypothesis are identified by an integer label. In the computation of constraint network configuration the pose values are updated using the constraints included by the most likely hypothesis, since the storage of all network configurations would be expensive. However, each edge corresponding to a different hypothesis stores the alternative values of the incident poses. In this way the network configuration can be computed for a different graph configuration, if the most likely hypothesis changes. Pruning criteria have been adopted to avoid the explosion of the hypotheses tracked by the robot.

II. PROBLEM FORMULATION

This section discusses the general formulation of the maximum likelihood (ML) approach and a solution algorithm based on Gauss-Seidel relaxation to solve the resulting equations. The SLAM problem is formulated as a graph, whose nodes correspond to the variables of the map and whose edges represent the constraints between pairs of these variables.

Graph-based SLAM can be expressed according to *feature based* or *delayed-state* representations [14]. In the following, the Gauss-Seidel relaxation is applied to the solution of the map in the delayed-state form. According to this formulation the map consists only of robot poses obtained by matching observations anchored to a local frame [5] or by marginalizing feature-based maps [15].

Then Let $\mathbf{p} = [p_1 \ \cdots \ p_n]^T$ be the vector of robot poses $p_i = [p_{i_x}, p_{i_y}, p_{i_\theta}]$. The notation ${}^u p_i$ will be used to note that pose p_i belongs to partition with numeric id u, when distributed relaxation is discussed. Otherwise, the partition label will be omitted. Let δ_{ij} and Ω_{ij} be respectively the mean and the information matrix of an observation of node j seen from node i. Let $f_{ij}(\mathbf{p})$ be a function that computes a zero noise observation according to the current configuration of the nodes j and i

$$f_{ij}(\mathbf{p}) = \begin{bmatrix} (p_{j_x} - p_{i_x}) \ \cos p_{i_\theta} + (p_{j_y} - p_{i_y}) \ \sin p_{i_\theta} \\ -(p_{j_x} - p_{i_x}) \ \sin p_{i_\theta} + (p_{j_y} - p_{i_y}) \ \cos p_{i_\theta} \\ p_{j_\theta} - p_{i_\theta} \end{bmatrix} \quad (1)$$

Thus, the error on constraint $\langle i, j \rangle$ is given by

$$e_{ij}(\mathbf{p}) \quad = \quad f_{ij}(\mathbf{p}) - \delta_{ij} \quad (2)$$

Let $\mathcal{E} = \{\langle i_1, j_1 \rangle, \ldots, \langle i_M, j_M \rangle\}$ be the set of pairs of indices for which a constraint $\delta_{i_m j_m}$ exists. Then the aim of ML approach is to minimize the negative log-likelihood or error function on the observation

$$\chi^2(\mathbf{p}) \quad = \quad \sum_{\langle i,j \rangle \in \mathcal{E}} e_{ij}^T(\mathbf{p}) \ \Omega_{ij} \ e_{ij}(\mathbf{p}) \quad (3)$$

Several numeric techniques have been proposed in order to find the minimum of $\chi^2(\mathbf{p})$, and many of them are based on fixed-point iteration. The likelihood function is usually approximated by a linear function and then the linearized system is solved. In particular, the observation functions $f_{ij}(\mathbf{p})$ are linearized around the current estimated value of the network configuration $\hat{\mathbf{p}}$

$$e_{ij}(\mathbf{p}) \approx f_{ij}(\hat{\mathbf{p}}) - \delta_{ij} + J_{ij}^i \Delta p_i + J_{ij}^j \Delta p_j \quad (4)$$

where J_{ij}^i and J_{ij}^j are the Jacobians of the observation function with respect to x_i and x_j evaluated in point $\hat{p}_i$ and $\hat{p}_j$. Since Eq. (1) only depends on poses i and j, there are no additional terms. Then, the resulting negative log-likelihood $\chi^2(\mathbf{p})$ is approximated by a quadratic function, whose minimum can be found by solving the linear system $A \ \mathbf{x} = b$, where A and b consists of block-elements

$$A_{ij} \quad = \quad \sum_{\langle i,h \rangle \in \hat{\mathcal{E}}} {J_{ih}^h}^T \ \Omega_{ih} \ J_{ih}^i + \sum_{\langle h,j \rangle \in \hat{\mathcal{E}}} {J_{hj}^h}^T \ \Omega_{hj} \ J_{hj}^j \quad (5)$$

$$b_i \quad = \quad \sum_{\langle i,h \rangle \in \hat{\mathcal{E}}} {J_{ih}^h}^T \ \Omega_{ih} \ (f_{ih}(\hat{\mathbf{p}}) - \delta_{ih}) + \\ + \quad \sum_{\langle h,i \rangle \in \hat{\mathcal{E}}} {J_{hi}^h}^T \ \Omega_{hi} \ (f_{hi}(\hat{\mathbf{p}}) - \delta_{hi}) \quad (6)$$

The linear system solution can be computed using elimination algorithm, Cholesky and other factorizations or relaxation methods. Gauss-Seidel relaxation has been adopted in this paper because it is easy to implement and it does not require back-substitution, even though there are methods with faster convergence. Indeed, back-substitution requires temporary variables for storing partial results and introduces stronger dependencies between state variables. In particular, the solutions of constraint networks corresponding to different hypotheses dramatically change.

III. MULTI-HYPOTHESIS CONSTRAINT NETWORK

The pose-graph representation illustrated in previous section has been adapted to support efficient storage of multi-hypothesis on a metric map. While the robot moves in the environment, new poses and constraints between poses are inserted in the map. The robot may visit or observe new or already visited locations and the graph building process has to settle whether loop closure occurs. Each hypothesis corresponds to a different outcome of data association history.

Since topological information is stored using limited memory space, a topological graph could be stored for each hypothesis as proposed in [12]. Conversely, an efficient management of multi-hypothesis metric constraint networks can be done by storing all the hypotheses in the same graph. The algorithm has been implemented according to the following properties.

- Given the current hypothesis h, each new hypothesis h' is defined by adding (or not adding) a new edge to the topology inherited from h. Thus, each hypothesis is associated to a single edge $\mathrm{edge}(h)$, but not viceversa since different hypotheses may be generated by inserting the same edge in different topologies. Each edge

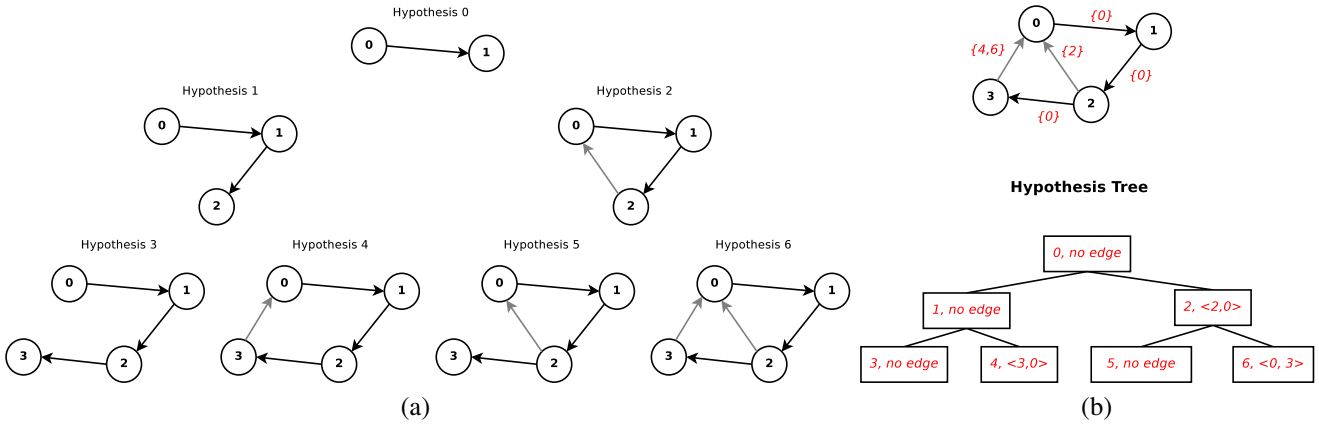

Fig. 1. An example of a network with multiple hypotheses (a). All the hypotheses can be stored in the edges of a single graph and in hypothesis tree (b). Each node of the hypothesis tree also has a pointer to the edge associated to the hypothesis, if there is one.

$\langle i, j \rangle$ stores the *id* of all the related hypotheses in set $hyp(\langle i, j \rangle)$.

- Whenever the robot moves, a new pose and a new edge representing the robot path are added to the graph even though the new pose is close to an already inserted pose. This choice may lead to a concentration of robot poses in the same region of the map [16], but it simplifies the hypotheses management since hypothesis are related only to edges. The edges of the path are labeled with a root hypothesis (with id equal to 0).
- The pose configuration of the constraint network is computed accordingly to the a single hypothesis $\bar{h}$ selected with a proper criterion. Still, each hypothesis h stores the values of the poses $h.p_i$ and $h.p_j$ connected by $\text{edge}(h) = \langle i, j \rangle$. Thus, if h will prevail over $\bar{h}$, the convergence to the new configuration is sped up by substituting $h.p_i$ and $h.p_j$.

A. Building the Hypothesis Tree

Loop closure hypotheses are kept in a hypothesis tree $\mathcal{H}$ similar to that in [9], [12]. Each branch of the tree represents the derivation of one hypothesis at a given time step k from another hypothesis at the previous step $k-1$. A level of the tree represents a different time step with one relevant exception: the root of the hypothesis tree labels all the edges representing robot trajectories. The path from the root to the current hypothesis has a role in the optimization step, since it idenfies the edges used in the computation of network configuration in current hypothesis. Figure 1 shows an example of constraint network with multiple hypotheses stored on a single graph.

Algorithm 1 illustrates how the update of graph and hypothesis tree is performed when a new measurement between current and previous pose is provided. First, the root hypothesis is recovered to label the new edge that belongs to the robot trajectory and is shared by all the hypotheses. Second, all the leaf hypotheses are visited and for each of them the loop closure candidates are found. The implementation of functions `findCandidateLoop`, which returns candidates poses for loop closure, and `estimateConstraint`, which estimates the pose displacement between current and candidate node, partly depends on the representation of local map in each node. While in topological maps hypotheses exhaustively covers all the possible topologies, `findCandidateLoop` selects candidates only among the nodes that are close to the current pose in order to keep the number of hypotheses small. The estimation of constraints corresponds to an alignment of local maps stored in the two nodes. Then, each new hypothesis and the related edge are added to the map and to the hypothesis tree.

Data: $\mathcal{G} = (\mathcal{N}, \mathcal{E})$: pose graph; $\mathcal{H}$: hypothesis list;
$z_k = (\mu_k, \Omega_k)$: measurement w.r.t. previous pose
Result: $\mathcal{G}$: updated pose graph; $\mathcal{H}$: updated hypothesis list
`/* add trajectory constraint common */`
`/* to all hypotheses */`
$rooth = \text{root}(\mathcal{H})$;
$p_{k-1} = \text{lastPose}(\mathcal{N})$;
$p_k = p_{k-1} \oplus \mu_k$;
$\mathcal{N} = \mathcal{N} \cup \{\text{node}(p_k)\}$;
$\mathcal{E} = \mathcal{E} \cup \{\text{edge}(\langle k-1, k \rangle, \mu_k, \Omega_k, rooth)\}$
`/* add and update hypotheses */`
foreach *hypothesis* $h \in \mathcal{H}_{leaf}$ **do**
  $\hat{h} = \text{createHypothesis}(\mathcal{H})$; $\text{parent}(\hat{h}) = h$;
  $\mathcal{H} = \mathcal{H} \cup \{\hat{h}\}$; $\mathcal{C} = \text{findCandidateLoop}(p_k)$;
  foreach *candidate* $n \in \mathcal{C}$ **do**
    $\hat{h} = \text{createHypothesis}(\mathcal{H})$;
    $\text{parent}(\hat{h}) = h$;
    $(\hat{\mu}, \hat{\Omega}) = \text{estimateConstraint}(k, n)$;
    $\mathcal{E} = \mathcal{E} \cup \{\text{edge}(\langle k, n \rangle, \hat{\mu}, \hat{\Omega}, \hat{h})\}$
    $\mathcal{H} = \mathcal{H} \cup \{\hat{h}\}$;
  end
end

Algorithm 1: Update of multi-hypothesis constraint network with a new measurement.

B. Pruning the Hypothesis Tree

The number of hypotheses contained in the hypothesis tree tends to explode due to the unbounded tree expansion illustrated above. To keep the tree size bounded, it is necessary to prune some branches of the tree. Several pruning criteria have been suggested in literature as shown

by the survey in [12]. Some are tuned on specific sensor or topological graph model (e.g. the degree test is based on the local topology of the Voronoi graph computed in vertices). Other techniques (like planarity and posterior probability tests) are more general and can be applied also to the proposed method. Futhermore, since the constraint network also contains metric information, the pairwise consistency test [17], which estimates the consistency of multiple edges according to the rigid transformation constraints, can also be applied to prune inconsistent hypotheses. In this work, we tried to avoid hypothesis tree pruning whenever possible by limiting the candidate edges for loop closure. Then, the number of hypotheses has been forcibly fixed to a certain bound `maxLeafHypotheses`.

IV. Solving Multi-hypothesis Network

In the previus section, the management of multi-hypothesis tree has been illustrated. The algorithm handling hypotheses is quite similar to topological map ones [12] except for storing all the graph topologies in the same graph. It would be convenient to share also metric configuration, i.e. the state vector of poses, between the hypotheses. Unfortunately, it would be inefficient to store and handle a network configuration for each hypothesis. Hence, the poses of the constraint network are computed according to a single hypothesis selected with maximum likelihood criterion. Partial metric information is stored within hypotheses in order to allow a fast convergence to the new configuration when the dominant hypothesis changes.

Let $\hat{h}$ be a leaf of hypothesis tree $\mathcal{H}$ and $\mathcal{P}_{\hat{h}}$ the path connecting $\hat{h}$ to the root. Set $\mathcal{P}_{\hat{h}}$ corresponds to all the ancestor hypotheses of $\hat{h}$. The function likelihood function ${}^{\hat{h}}\chi^2(\mathbf{p})$ associated to this hypothesis would be

$$ {}^{\hat{h}}\chi^2(\mathbf{p}) = \sum_{\langle i,j\rangle \in \mathrm{edge}(\mathcal{P}_{\hat{h}})} e_{ij}^T(\mathbf{p})\ \Omega_{ij}\ e_{ij}(\mathbf{p}) \qquad (7) $$

where $\mathrm{edge}(\mathcal{P}_{\hat{h}})$ is the collection of the constraints associated to the hypotheses $h \in \mathcal{P}_{\hat{h}}$ or, formally,

$$ \mathrm{edge}(\mathcal{P}_{\hat{h}}) = \left\{ e \in \mathcal{E} : \exists h \in \mathcal{P}_{\hat{h}}, e = \mathrm{edge}(h) \right\} \qquad (8) $$

Therefore, the edges associated to a hypothesis not beloging to the path do not contribute to the configuration of the network. However, each hypothesis h associated to $\mathrm{edge}(h) = \langle i,j\rangle$ stores the values of the poses $h.p_i$ and $h.p_j$ computed according to such hypothesis. The constraint linearization for a given hypothesis is performed accoding to equation (4), but the values of residual $f_{ij}(\hat{\mathbf{p}}) - \delta_{ij}$ and of Jacobians J_{ij}^i and J_{ij}^j are computed using the poses stored in the hypotheses. Hence, for each hypothesis h the block-matrix elements ${}^hA_{ij}$ and block-vector elements ${}^hb_i$ of the linearized system are given by

$$ \begin{aligned} {}^hA_{ij} &= \sum_{\langle i,k\rangle \in \mathrm{edge}(\mathcal{P}_h)} {J_{ik}^k}^T\ \Omega_{ik}\ J_{ik}^i + \\ &+ \sum_{\langle k,j\rangle \in \mathrm{edge}(\mathcal{P}_h)} {J_{kj}^k}^T\ \Omega_{kj}\ J_{kj}^j \qquad (9) \\ {}^hb_i &= \sum_{\langle i,k\rangle \in \mathrm{edge}(\mathcal{P}_h)} {J_{ik}^k}^T\ \Omega_{ik}\ (f_{ik}({}^h\hat{\mathbf{p}}) - \delta_{ik}) + \\ &+ \sum_{\langle k,i\rangle \in \mathrm{edge}(\mathcal{P}_h)} {J_{ki}^k}^T\ \Omega_{ki}\ (f_{ki}({}^h\hat{\mathbf{p}}) - \delta_{ki}) \qquad (10) \end{aligned} $$

Solution of network is achieved by applying Gauss-Seidel relaxation to all the nodes and to all the poses in the hypotheses. In particular, the expression of a single iteration for pose p_i given the hypothesis h is

$$ \begin{aligned} {}^h\Delta p_i &= {}^hA_{ii}^{-1}\left({}^hb_i - \sum_{i\neq j} {}^hA_{ij}\,{}^h\Delta p_j \right) \qquad (11) \\ &= {}^hD_i^{-1}\,{}^h\alpha_i \qquad (12) \end{aligned} $$

where the values of ${}^hD_i$ and of ${}^h\alpha_i$ are computed by accomulating the contributions of the incident edges that belong to $\mathcal{P}_h$. Consider the example of Figure 1. In the case of hypotheses 3 and 5 the adjustment of pose p_3 is performed using only edge $\langle 2,3\rangle$ (the only incident edge on paths $\mathcal{P}$ $\mathcal{P}_4$), while for hypothesis 4 and 6 the updated value depends both on $\langle 2,3\rangle$ and $\langle 3,0\rangle$. The difference between hypotheses 3 and 4 lies only in the value of the poses p_0 and p_3 used in the linearization of constraint $\langle 3,0\rangle$.

Observe that the values of ${}^hD_i$ and ${}^h\alpha_i$ is given by the *sum* of the independent contribution of the edges in $\mathcal{P}_h$ as

$$ \begin{aligned} {}^hD_i &= J_e^{iT}\Omega_e J_e^i + {}^{\bar{h}}D_i \qquad (13) \\ {}^h\alpha_i &= J_e^{iT}\Omega_e((f_e(\mathbf{p}) - \delta_e) - J_e^j h.\Delta p_j) + \\ &+ {}^{\bar{h}}\alpha_i \qquad (14) \end{aligned} $$

where $e = \mathrm{edge}(h) = \langle i,j\rangle$ or $e = \mathrm{edge}(h) = \langle j,i\rangle$ and $\bar{h} = parent(h)$. Equation 14 can be recursively expanded until root hypothesis is reached. Thus, ${}^hD_i$ and ${}^h\alpha_i$ can be computed by storing the partial contribution of each hypothesis in the hypothesis and then by accumultating the sum in a depth-first visit from the root to the leaves of the hypothesis tree. Algorithm 2 illustrate the whole algorithm.

V. Results

The proposed multi-hypothesis constraints network has been assessed using both simulation data and real data dataset. The algorithm has been implemented and the application incrementally inserts nodes and edges in the graph. In particular, two experiments have been performed, one in a simulated environment (SIM) and the other from commonly used dataset ACES building on UT Austin campus (ACES). Figure 2(a) shows the robot path related to the simulated exploration of an $20 \times 10\ m$ office-like environment with Player-Stage. Data have been collected by adding Gaussian noise with respectively $\sigma_{pos} = 0.01\ m$ and $\sigma_{pos} = 0.05^\circ$ every $0.25\ m$. The same amount of noise has been applied to every measurement scaled with respect to distance. Both networks SIM and ACES are built by adding one node at a time, the edge connecting the current pose to the previous one and eventually candidate edges representing possible closures. After the network update,Gauss-Seidel relaxation has been performed. The resulting constraints network for experiment SIM and experiment ACES referred to the dominant hypothesis are depicted respectively in Figure 2(b) and Figure 3. Since measurement likelihood is difficult to obtain, the dominant hypothesis has been selected as the hypothesis

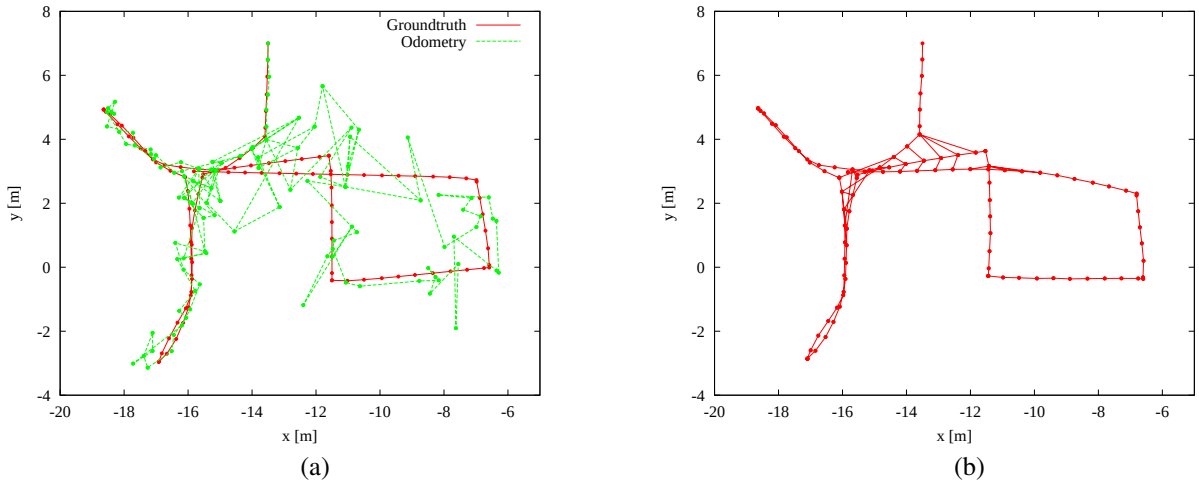

Fig. 2. The groundtruth and the noisy odometry of robot path in simulated experiment SIM (a) and the corresponding adjusted constraint network (b).

Data: $\mathcal{G} = (\mathcal{N}, \mathcal{E})$: pose graph; $\mathcal{H}$: hypothesis list; $\bar{h}$: current selected hypothesis
Result: $\mathcal{G}$: updated pose graph
foreach *node* $n \in \mathcal{N}$ **do**
  $\mathcal{L} = \emptyset$;
  /* constraints ingoing into node n */
  foreach $\langle i, n \rangle \in \hat{\mathcal{E}}$ **do**
    foreach $h \in hyp(\langle i, n \rangle)$ **do**
      computes residual $res = f_{in}(\mathbf{p}) - \delta_{in}$
      and Jacobians J^i_{in} and J^n_{in} in $h.p_i, h.p_n$);
      ${}^h D_n = J^{n\,T}_{in} \Omega_{in} J^n_{in}$;
      ${}^h \alpha_n = J^{n\,T}_{in} \Omega_{in} (J^n_{in} res - J^i_{in} h.\Delta p_i)$;
      $\mathcal{L} = \mathcal{L} \cup \{h\}$;
    end
  end
  /* constraints outgoing from node n */
  foreach $\langle n, i \rangle \in \hat{\mathcal{E}}$ **do**
    foreach $h \in hyp(\langle n, i \rangle)$ **do**
      computes residual $f_{ni}(\mathbf{p}) - \delta_{ni}$
      and Jacobians J^n_{ni} and J^i_{ni} in $h.p_n, h.p_i$);
      ${}^h D_n = J^{n\,T}_{ni} \Omega_{ni} J^n_{ni}$;
      ${}^h \alpha_n = J^{n\,T}_{ni} \Omega_{ni} (J^n_{ni} res - J^i_{ni} h.\Delta p_i)$;
      $\mathcal{L} = \mathcal{L} \cup \{h\}$;
    end
  end
  /* propagation of partial terms D and α */
  sort $\mathcal{L}$ according to the depth in tree;
  foreach $h \in \mathcal{L}$ **do**
    foreach $h' \in \mathcal{P}_h$ **do**
      ${}^h D_n = {}^h D_n + {}^{h'} D_n$;
      ${}^h \alpha_n = {}^h \alpha_n + {}^{h'} \alpha_n$;
    end
    $h.\Delta p_n = {}^h D_n{}^{-1}\, {}^h \alpha_n$; ${}^h D_n = 0$;
    ${}^h \alpha_n = 0$;
    if $h \in \mathcal{P}_{\bar{h}}$ **then**
      $\Delta p_n = h.\Delta p_n$;
    end
  end
end
/* update $p_n = p_n + \Delta p_n$ for each $n \in \mathcal{N}$ */
/* update $h.p_i = p_i + \Delta p_i$ for each $h \in \mathcal{H}$ */

Algorithm 2: Gauss-Seidel iteration on multi-hypothesis constraint network.

with less average error per loop constraint. It is apparent that in both cases the resulting maps are consistent.

Figures 4 and 5 show the number of hypotheses and of constraints with respect to the number of nodes included in the graph respectively for SIM and ACES. The final graphs for SIM and ACES contain respectively 108 nodes and 137 edges and 648 nodes and 1730 edges. However, the size of hypothesis tree is respectively equal to 2907 and 27382 applying the pruning criterion described in section III. Of course, the number of hypotheses deeply affects the time required to perform a Gauss-Seidel iteration. The time required by an iteration is 4 s for the final ACES graph, whereas it is 25 ms for the final SIM model on an Intel Core Dual processor T2300 with 1 GB memory. The computational load is mostly due to the propagation of partial results D and α in Algorithm 2), whose optimization should be investigated in future work. It should be remarked that the number of hypotheses that can be managed by the proposed algorithm is comparable to the number of hypotheses managed in [12] on a topological map.

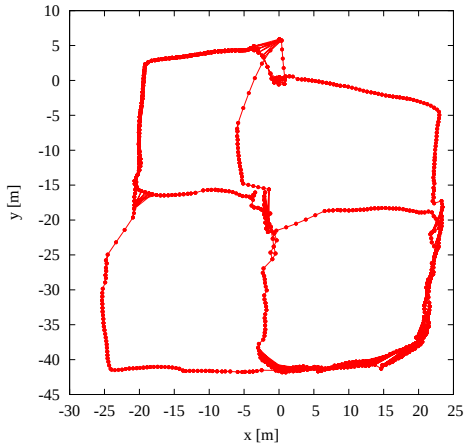

Fig. 3. The solved constraint network corresponding to dataset ACES.

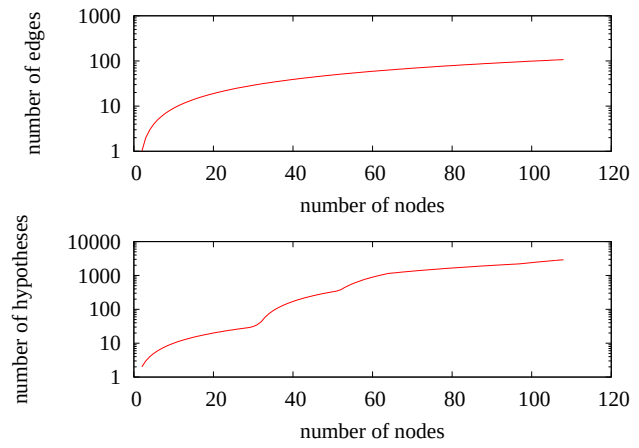

Fig. 4. Figure shows the number of edges (top) and the number of hypotheses (bottom) with respect to the number of nodes for SIM.

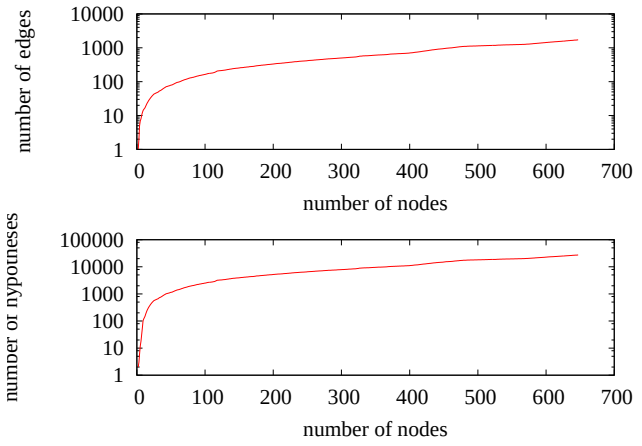

Fig. 5. Figure shows the number of edges (top) and the number of hypotheses (bottom) with respect to the number of nodes for dataset ACES.

VI. CONCLUSION

In this paper, we have proposed a multi-hypothesis constraint network algorithm that tracks multiple map topologies. Such representation also encodes metric information since the map is stored in a graph consisting of poses and constraints. Each loop closure hypothesis is directly associated to the edge of the graph that possibly closes the loop. Hypotheses are stored in a hypothesis tree that represents the derivation of each hypothesis from previous hypotheses. After a new pose is added to the constraint network and candidate edges for associations are found, the hypothesis tree is expanded to include all the outcomes of data association. A naive pruning technique has been used to bound the number of hypotheses.

The pose values are stored in the nodes of the graph and are computed according to the dominant hypothesis, but each hypothesis keeps the values of the two poses that are connected according to such hypothesis. Gauss-Seidel relaxation is applied to compute the pose value for each hypothesis. Using relaxation algorithm the pose of a node is estimated by summing the contributions of each incident edges. Thus, the value of the pose stored in each hypothesis can be updated by properly distributing such contributions. Experiments assess the consistency of the proposed approach, even though computation time significantly increases with the number of hypotheses.

In future work, we expect to introduce a more efficient pruning technique to reduce the hypotheses and to improve the efficiency of the relaxation algorithm. The propagation of partial residuals through the hypothesis tree is a time consuming operation that could be sped up by keeping proper references at each node.

VII. ACKNOWLEDGMENTS

This research is partially supported by laboratory AERTECH of Regione Emilia-Romagna, Italy.

REFERENCES

[1] J. Neira and J. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics*, vol. 17, no. 6, pp. 890–897, 2001.

[2] H. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[3] I. Cox and M. Miller, "On finding rank assignment with application to multi-target tracking and motion correspondence," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 32, pp. 486–489, 1995.

[4] T. Bailey, "Mobile robot localization and mapping in extensive outdoor environments," Ph.D. dissertation, University of Sidney, 2002.

[5] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Journal of Autonomous Robots*, vol. 4, pp. 333–349, 1997.

[6] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1999, pp. 318–325.

[7] P. Newman and K. Ho, "SLAM - Loop Closing with Visually Salient Features," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.

[8] K. Ho and P. Newman, "Detecting loop closure with scene sequences," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 261–286, September 2007.

[9] G. Dudek, P. Freedman, and S. Hadjres, "Using local information in a non-local way for mapping graph-like worlds," in *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 1993.

[10] D. Haehnel, S. Thrun, B. Wegbreit, and W. Burgard, "Towards lazy data association in SLAM," in *Proc. of the Int. Symposium of Robotics Research (ISRR)*, 2003.

[11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.

[12] S. Tully, G. Kantor, H. Choset, and F. Werner, "A multi-hypothesis topological slam approach for loop closing on edge ordered graphs," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[13] D. Lodi Rizzini and S. Caselli, "A distributed maximum likelihood algorithm for multi-robot mapping," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[14] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005, pp. 2428–2435.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[16] H. Kretzschmar, G. Grisetti, and C. Stachniss, "Life-long map learning for graph-based simultaneous localization and mapping," *Kuenstliche Intelligenz (KI)*, vol. 24, no. 3, pp. 199–206, 2010.

[17] E. Olson, "Robust and efficient robotic mapping," Ph.D. dissertation, Massachusetts Institute of Technology, 2008.